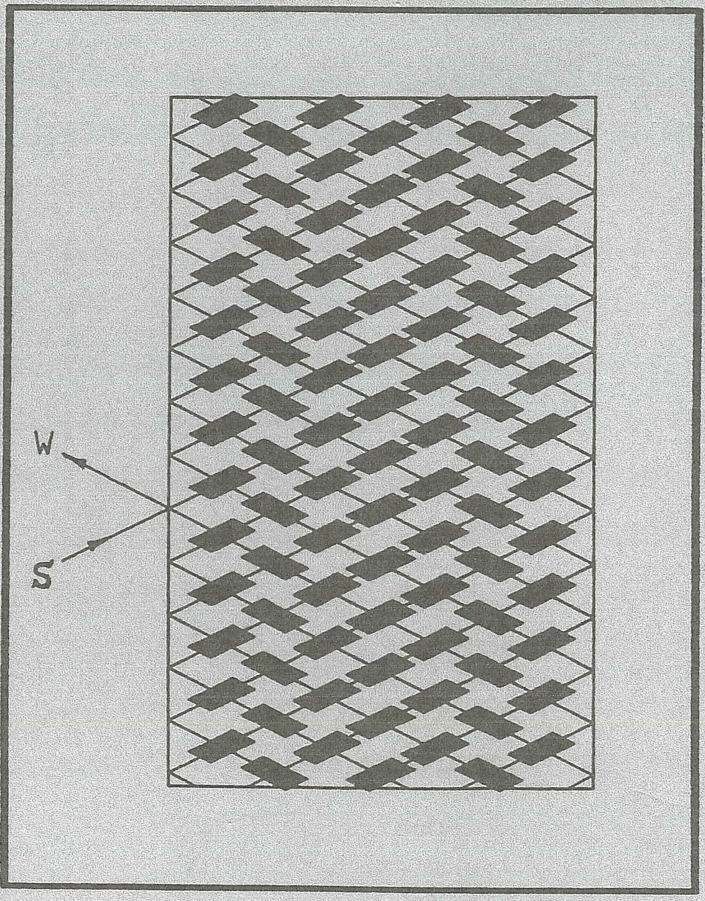
THE BRAIDING OF COLUMN-CODED REGULAR KNOTS



A.G. Schaake
J.C. Turner

7

Department of Mathematics and Statistics.
University of Waikato.

Topics in Braiding Theory and Practice

Pamphlet No.7

ISSN. 1170-7003

OTHER PAMPHLETS IN THIS SERIES :

Pamphlet No. 1 .... **Introducing Grid-diagrams in Braiding**
Pamphlet No. 2 .... **Edge Lacing --- The Double Cordovan Stitch**
Pamphlet No. 3 .... **Braiding Application --- Horse Halter**
Pamphlet No. 4 .... **The Regular Knot Tree and Enlargement Processes**
Pamphlet No. 5 .... **An Introduction to Flat Braids**
Pamphlet No. 6 .... **An Introduction to Evolution Processes (Part 1)**

These pamphlets may be obtained from :

Dr. J.C. Turner,
Department of Mathematics and Statistics,
University of Waikato, Private Bag 3105,
Hamilton,
New Zealand.

OR FROM

Mr. D.Pawson,
Footrope Knots,
501 Wherstead Road,
Ipswich, Suffolk, IP2 8LL,
England.

Copyright © -- A.G. Schaake, J.C. Turner.

All rights reserved.

Free use of the material in this pamphlet is encouraged, provided that the source is fully acknowledged.

March, 1992.

The Braiding of Column-coded Regular Knots

A.G. Schaake
J.C. Turner

ISBN. 0-908830-11-4

Abstract:

A general method for obtaining the information which enables a braider to braid any Column-coded Regular Knot is fully discussed, together with the necessary conventions used for Regular Knots. Refinements, such as an aid in carrying out a continual check on the braiding progress; and the construction of a Column-coded Regular Knot by starting at the midpoint of its string-length, are dealt with. Finally, braiding to a weaving-pattern is briefly treated.

For copies, write to:

Dr. J.C. Turner,
Department of Mathematics and Statistics,
University of Waikato, Private Bag 3105,
Hamilton,
New Zealand.

CONTENTS

page

INTRODUCTION 1

COLUMN-CODED REGULAR KNOTS 2

SOME BRAIDING MATHEMATICS 4

SOME CONVENTIONS 7

A GENERAL BRAIDING METHOD 12

REFINEMENTS 26

BRAIDING BY WEAVING-PATTERN 32

REFERENCES 37

Introduction

In this pamphlet we shall discuss a method for braiding Column-coded Regular Knots, which is easy to master by those readers who do not possess any mathematical knowledge. The method is not only easy to learn, but also it is one of the most general and comprehensive ones for obtaining all the information a braider requires to construct any Column-coded Regular Knot. Although it has been fully described by us in an earlier publication (see Ref. [2] pages 21 – 23), it appears that due to the brevity and mathematical nature of the discussion there, most braiders have unfortunately been unable to apply it to their craft. For this reason, and since the method is a very comprehensive, powerful and easy one, we feel justified in devoting a non-mathematical pamphlet to it.

Before discussing the method in detail, we shall lead the reader slowly through the concepts of Regular Knots, grid-diagrams, Column-coding, and some other general conventions used by us in all our publications on braiding. Since braids are natural phenomena, they are associated with what is called *"modular arithmetic"*. This type of arithmetic can justifiably be called "nature's arithmetic", since it is associated with all natural cyclic processes. Everybody is to some extent familiar with it, despite the lack of attention it receives in our educational system. We shall discuss the essentials of this type of arithmetic, as far as required in braiding, fully in the Section *"Some Braiding Mathematics"*.

Although the braiding method we shall describe is suitable for braiding to a weaving-pattern of a Column-coded Regular Knot (a method still used by many braiders), we do not recommend such a procedure, since it is too restrictive with respect to knot sizes. It is advisable to braid either directly to an algorithm-diagram, or to an algorithm-table which can be derived from it. Such an algorithm-table consists of consecutive half-cycle braiding algorithms. Only by using one of these two procedures can we braid efficiently any possible parts/bights combination; an essential requirement for braiding with any available string-width. We shall explain with the aid of several examples the method for obtaining the algorithm-diagram and its associated algorithm-table. The algorithm-diagram is the focal item in the braiding method discussed in this pamphlet. It also provides information which enables us to introduce refinements in the construction process, such as keeping a continual check on the correctness of the braiding progress, a very handy, and for large knots essential, procedure in braiding efficiently.

Although in this pamphlet we limit our discussion to Column-coded Regular Knots, the method dealt with can readily be adapted for Row-coded Regular Knots. If sufficient interest is expressed in this, we will produce a further pamphlet, dealing with Row-coded Regular Knots. We would appreciate suggestions and comments from readers, to help us in producing the topics in our publications in the best possible way.

Column-Coded Regular Knots

Each of the Figures 1, 2 and 3 represents a column-coded *Regular* Flat Braid of seven strings and five bights. A *Regular* Flat Braid is a flat braid having all its left-hand bights on a single straight line and at the same time all its right-hand bights on another parallel single straight line. It consists of two sets of parallel string-segments running from one bight-boundary to the other.

Imagine that behind any of the braids is placed a horizontal cylinder and that we wrap the braid backwards around the cylinder, joining the top number 1 to the bottom number 1, the top number 2 to the bottom number 2, the top number 3 to the bottom number 3, ..., and finally the top number 7 to the bottom number 7. The result is a Regular Cylindrical Braid.

If we now follow any chosen string round and around the cylinder, we shall find that after seven complete revolutions we return to our starting point. Thus the Regular Cylindrical Braid can be made from one single string. Such *single* string braids, which can be thought of as being endless Regular Flat Braids, are called **Regular Knots**.

By reversing the process, we can cut a Regular Knot along a cylinder-generating line which lies between two *adjacent rows* of string-intersections, and then unroll it into a flat plane in *front* of the cylinder. A simple projection onto the plane then provides the *grid-diagram* of the knot.

The grid-diagram consists of the following two parts:

(1) The **string-run**. This is the way the string is laid down upon the cylinder, without regard to the inter-weaving required.

(2) The **coding**. This is the way in which the desired weave-pattern is achieved, as the string is laid down.

In the Figs. 1, 2, 3 the central diagrams represent the string-run, and the right-hand diagrams represent the string-run and coding (the complete grid-diagram).

In these grid-diagrams we can draw vertical lines through the string-intersections. Such vertical lines are called intersection-columns. A Regular Knot is Column-coded when each intersection-column has the same coding. That is, when all intersections in a given column have the same coding. Note that different columns may be coded differently. Hence Regular Knots with grid-diagrams as in Figs. 1, 2 and 3 are Column-coded Regular Knots.

The distance between two adjacent vertical lines represents one part.

Before discussing some different methods by which these knots can be braided, we shall first introduce the reader to a little mathematics that serves as a useful tool in these methods.

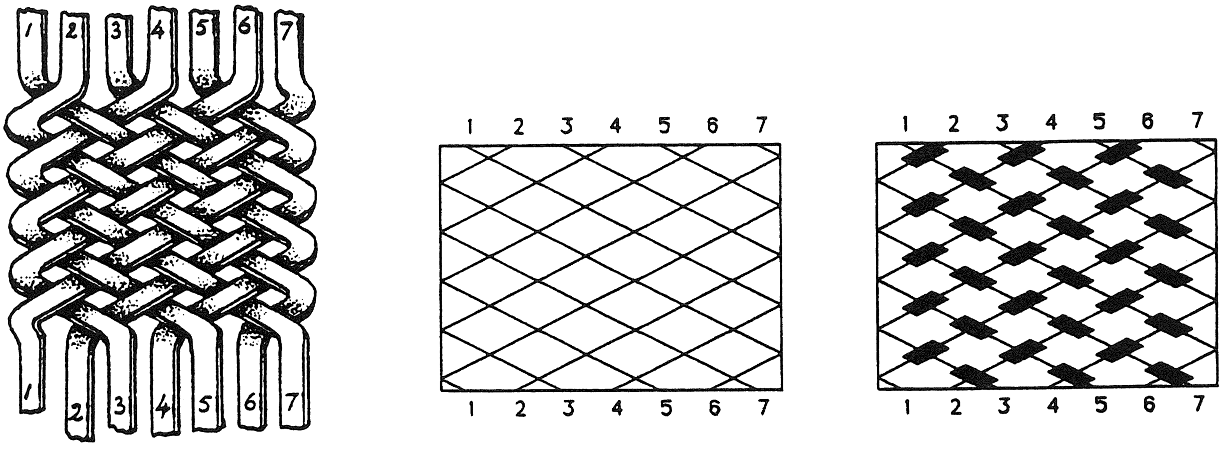


Fig. 1 — Turk's Head Coding.

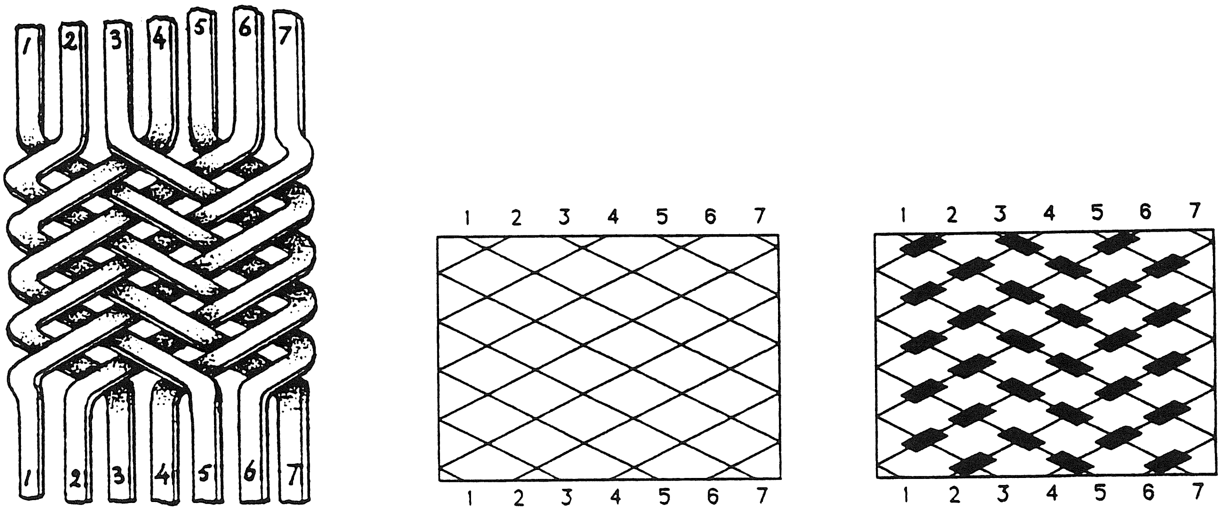


Fig. 2 — Two-pass Headhunter's Coding.

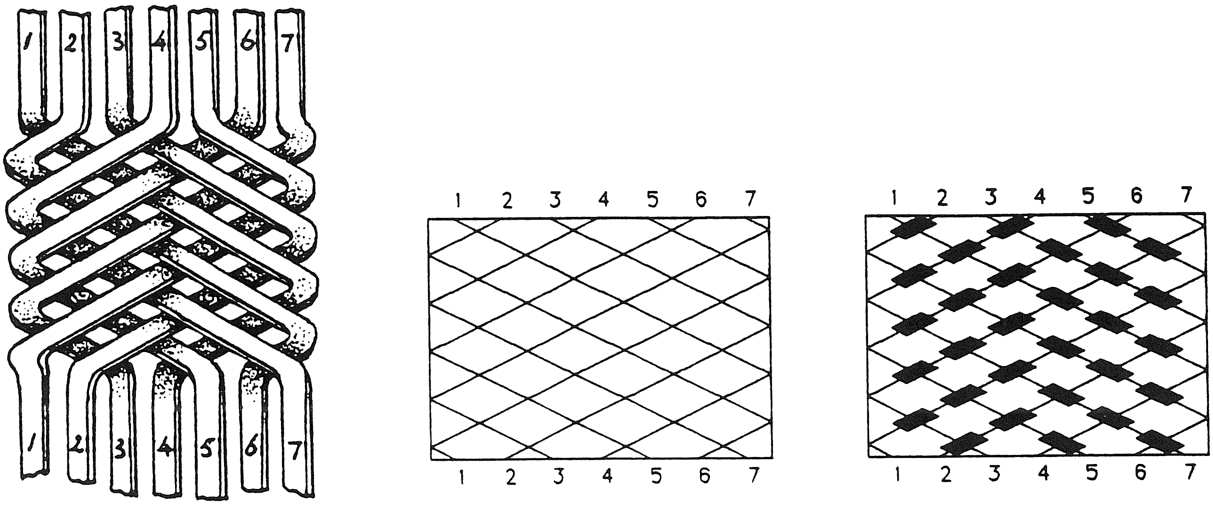
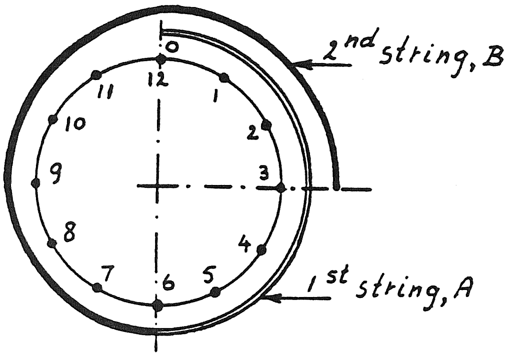


Fig. 3 — Three-pass Ring Knot Coding.

Some Braiding Mathematics

When constructing cylindrical braids, a string is passed round and around a cylinder, forming **half-cycles** (from left-hand bights to right-hand bights and vice-versa) in a string-run as it goes. Each time it passes round the cylinder, the newly formed half-cycles may overlay those which were formed previously. In order to study relative bight and string-crossing positions on a cylinder, it is natural to use what is called **modular arithmetic** (or clock arithmetic). It is in fact nature's cyclic arithmetic.

Suppose we mark twelve equi-distant points around a section of a cylinder (to use a "12-hour clock" example), and wrap two portions of string (say A and B) round it, in a clockwise direction.



Let the length of portion A be equal to 6 units, and the length of portion B be equal to 9 units.

The strings are joined end-to-end, so their total length is 15 units.

The string begins at position 0 (equivalent to position 12) and ends at position 3.

Thus, by "ordinary" arithmetic we would obtain $A + B = 15$; however by "clock arithmetic" we obtain $A + B = 3$.

In effect, every time we pass completely round the cylinder we discard 12 units from our sum, and start counting from 0 again. This is called modulo-12 addition.

Our notation for this is $|A + B|_{12}$. We use the notation throughout all our books on Braiding.

The symbols indicate that the ordinary sum of the lengths A and B has to be reduced by as many multiples of 12 as it contains; the "answer" is the smallest possible positive or zero remainder. In general, we refer to this value simply as the remainder.

Sometimes in our braiding applications, when it happens that the remainder is $r = 0$, we find it necessary to use the modulus-value (i.e. 12) instead of 0 (refer to the clock diagram, where it is seen that 0 and 12 are equivalent, and therefore interchangeable).

In everyday life, for example, we never say that the time is 0 o'clock, but always 12 o'clock.

Examples :

$|3 + 4|_{12} = |7|_{12} = 7$; since $7 \div 12$ leaves 7 as remainder.

$|9 + 10|_{12} = |19|_{12} = 7$; since $19 \div 12$ leaves 7 as remainder.

$|11 + 5|_{12} = |16|_{12} = 4$; since $16 \div 12$ leaves 4 as remainder.

$|6 + 6|_{12} = |12|_{12} = 0$ (sometimes written as 12).

The modulus of a negative integer can be compared with wrapping the string around the cylinder in the anti-clockwise direction. For example, wrapping the strings $A+B=15$ anti-clockwise on the modulo-12 clock takes us to the point 9 on the clock. Hence we can write:

$$|-15|_{12} = |-3|_{12} = 9 .$$

The easiest way to work this answer out is to take the nearest multiple of 12 above 15, which is 24, then subtract 15 from 24 which gives the answer 9.

Further examples are:

$$|5-9|_{12} = |-4|_{12} = 8 ; \quad \text{work out } (12-4), \text{ which gives } 8 .$$
$$|6-11|_{12} = |-5|_{12} = 7 ; \quad \text{work out } (12-5), \text{ which gives } 7 .$$
$$|3-22|_{12} = |-19|_{12} = |-7|_{12} = 5 ; \quad \text{work out } (24-19), \text{ which gives } 5 .$$
$$|7+2-10|_{12} = |-1|_{12} = 11 ; \quad \text{work out } (12-1), \text{ which gives } 11 .$$

Numbers other than 12 may be used as a modulus. Once the modulus is given, a complete system of addition, subtraction, multiplication and division may be defined, in a similar fashion to that of ordinary arithmetic.

Examples:

$$|9|_3 = 0 \text{ (sometimes written as 3)} .$$
$$|21|_5 = 1 ; \quad \text{since } 21 \div 5 \text{ leaves 1 as remainder..}$$
$$|8+15|_6 = |23|_6 = |5|_6 = 5 .$$
$$|-7|_4 = |-3|_4 = 1 ; \quad \text{work out } (8-7) \text{ or } (4-3), \text{ which gives } 1 .$$
$$\big||8|_6 + |15|_6\big|_4 = |2+3|_4 = |5|_4 = |1|_4 = 1 .$$

In general notation, suppose the modulus is p. Then if N is any integer, it can be expressed in the form:

$$N = m \cdot p + r ,$$

where m is the number of multiples of p which causes the remainder r to be a number in the interval $0 \leq r < p$. The symbol "$\leq$" stands for "less or equal than", and the symbol "$<$" stands for "less than".

Note that m may be any integer; that is, m may be any whole number, positive, negative or zero.

Our general notation for the modulus operation is:

$$|N|_p = r .$$

The following Rules are associated with modular arithmetic. We shall not use them in this pamphlet, but some readers may be interested in them, and care to make up examples to demonstrate the truth of each one.

Rules :

(1) If $|A|_p = |B|_p$, and $|C|_p = |D|_p$, then :

$$|A + C|_p = |B + D|_p \,,$$
$$|A - C|_p = |B - D|_p \,,$$
$$|A \cdot C|_p = |B \cdot D|_p \,.$$

(2) $|A + B|_p = \Big| |A|_p + |B|_p \Big|_p$.

(3) $|n \cdot A|_{n \cdot p} = n \cdot |A|_p$.

(4) If $|A|_p = |B|_p$, we may **not** deduce that $A = B$; we can, however, deduce that $|A - B|_p = 0$. Conversely, $|A - B|_p = 0$ implies that $|A|_p = |B|_p$.

(5) If $|k \cdot A|_p = |k \cdot B|_p$, then :

$|A|_{\frac{p}{d}} = |B|_{\frac{p}{d}}$, where g.c.d.$(k, p) = d$.

"g.c.d.(k, p)" stands for: "the greatest common divisor of k and p".

Sometimes it is necessary to calculate the **absolute value** of a number. It is the *positive value* of a number. The symbol for the absolute value of a number is a vertical bar on each side of the number.

Examples :

$$|8| = 8 \,.$$
$$|0| = 0 \,.$$
$$|-4| = 4 \,.$$
$$|17 + 5| = |22| = 22 \,.$$
$$|3 - 26| = |-23| = 23 \,.$$
$$|-4 \cdot 3| = |-12| = 12 \,.$$
$$\left|\frac{-2 \cdot 8 + 1}{5}\right| = \left|\frac{-16 + 1}{5}\right| = \left|\frac{-15}{5}\right| = |-3| = 3 \,.$$

Some Conventions

To enable the efficient use of grid-diagrams, we have to follow some general conventions. Although opposing half-cycles intersect each other at about 90° , there is no readily available graph-paper with grid-lines at 45° to the short and long edges of the paper. There is, however, an *isometric* grid-paper available in which the main grid-lines intersect each other at 120° ; these lines make angles of 30° with the long side of the paper. Furthermore this graph-paper contains lines, running through the intersection points of the main grid, perpendicular to the long side of the paper, which makes this graph-paper ideally suited for the drawing of grid-diagrams. Fig. 4 gives a sample of the actual grid.

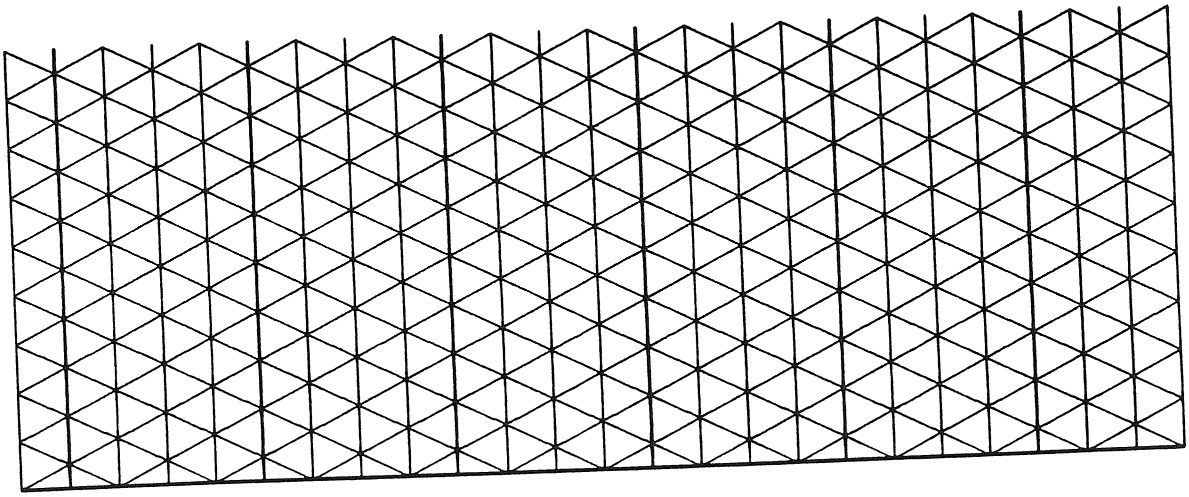


Fig. 4 — Isometric grid-paper.

This isometric graph-paper can be ordered from any stationary outlet, and is available in New Zealand in two sizes:

(1) Foolscap size; 50 sheet pad, No. 0810 Isometric Grid, Gormack Graph Papers : Christchurch N.Z.

(2) A 3 size; 50 sheet pad, No. L110X Isometric Grid, Gormack Graph Papers : Christchurch N.Z.

We have mentioned earlier how we obtain a grid-diagram from an actual Regular Knot, by an imaginary cut along a cylinder-generating line between two *adjacent rows* of string-intersections, and then unrolling it into a flat plane in *front* of the cylinder. A subsequent projection onto the plane provides the *grid-diagram* of the knot.

This imaginary cutting-line can be represented on the graph-paper in two ways: either as on the left-hand side of Fig. 5, or as on the right-hand side of Fig. 5.

Since we unrolled the "cut" knot, this imaginary cutting-line appears twice in the grid-diagram: at the top and at the bottom. The left-hand bight-boundary is the vertical bounding line at the left-hand side of the grid-diagram, and the right-hand bight-boundary is the vertical bounding line at the right-hand side of the grid-diagram. We mentioned earlier that the distance between two adjacent vertical lines represents one part, hence the horizontal "distance" between the two vertical bounding lines is equal to the number of parts of the Regular Knot.

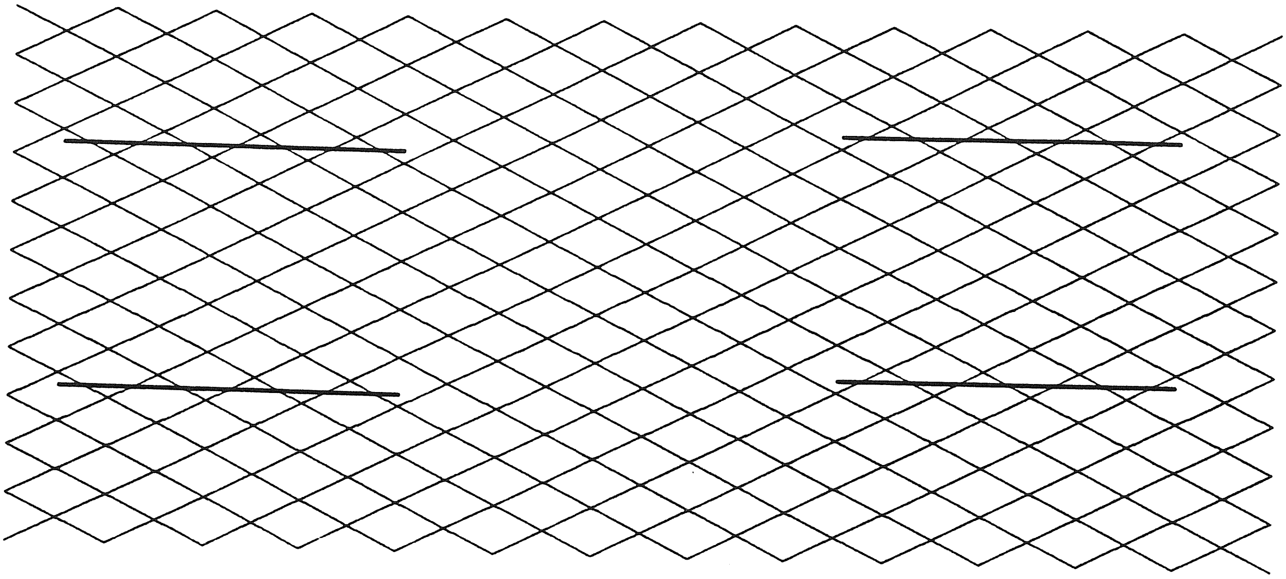


Fig. 5 — The imaginary cutting-line.

The bight positions are indicated by the intersection points of the grid-lines on these left-hand and right-hand vertical bounding lines. Consequently the "distance" between the top horizontal line and the bottom horizontal line of the grid-diagram is equal to the number of bights of the knot. Fig. 6 gives these bounding lines for a 7-part/5-bight Regular Knot for both methods of representation. It is immaterial which of these two methods is used.

> *It should be noted that although the imaginary cut could be made through a row of intersection-points on a cylinder-generating line, this should* **not** *be done, in order to avoid confusion in further uses of grid-diagrams in other braiding applications.*

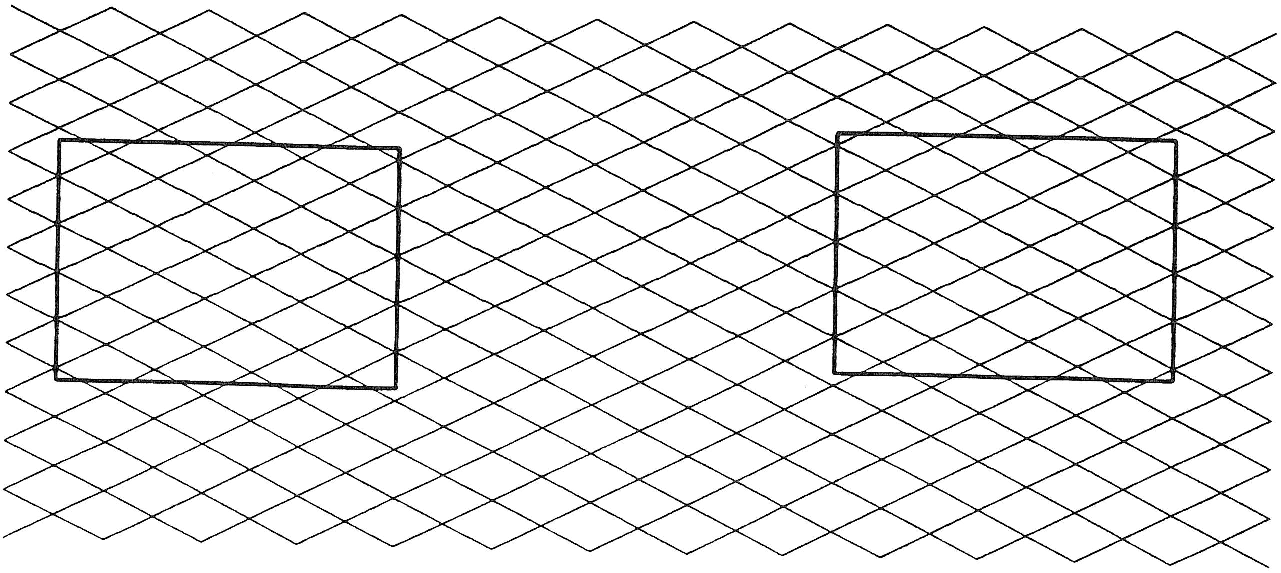


Fig. 6 — The outlines of a 7-part/5-bight Regular Knot.

The string-run of the Regular Knot is now represented by the slanting grid-lines within the horizontal and vertical bounding lines of the diagram. Fig. 7 gives the string-runs for a 7-part/5-bight Regular Knot for both methods of representation.

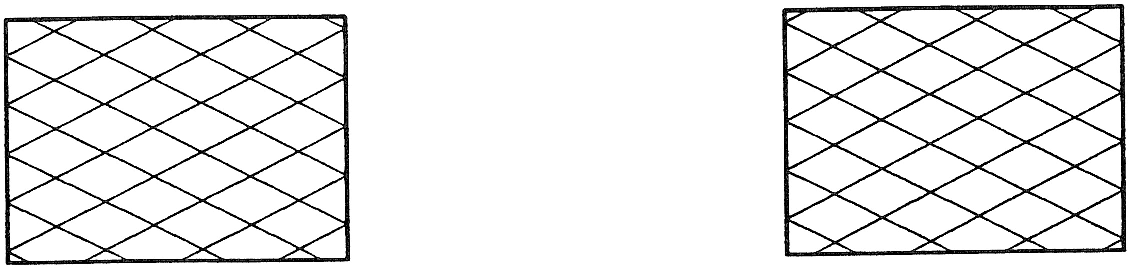


Fig. 7 — The string-run of a 7-part/5-bight Regular Knot.

All the vertical lines of the grid-diagram are numbered from *left to right* **and** from *right to left*. In each case we start with the appropriate bight-boundary, which receives the number " 0 "; hence the last vertical line to be numbered is the other bight-boundary, which consequently receives a number that is equal to the number of parts of the Regular Knot. Fig. 8 gives these numberings for a 7-part/5-bight Regular Knot for both methods of representation.

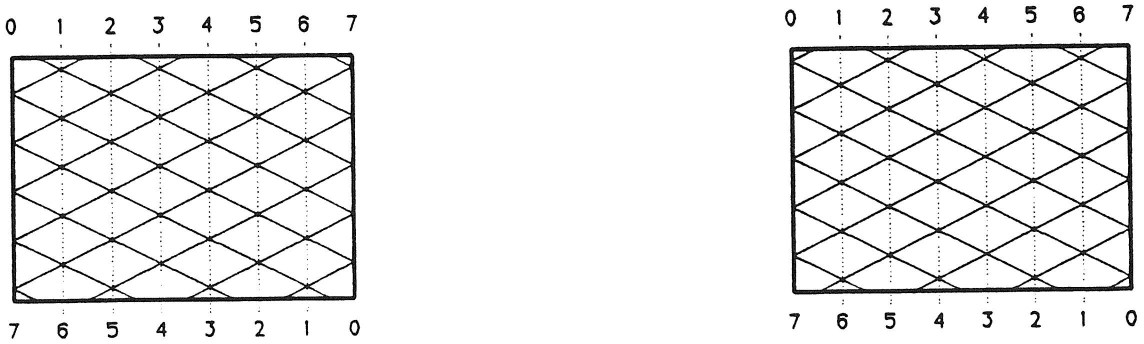
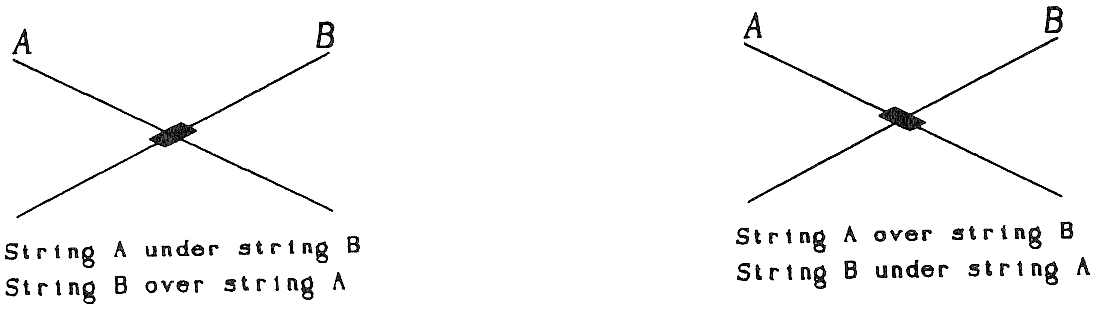


Fig. 8 — The numbering of columns.

The coding of a knot is represented by thick line-segments placed on crossings; they indicate the type of crossing which takes place at string "intersections". The convention we use in all our publications is presented in Fig. 9.



String A under string B
String B over string A

String A over string B
String B under string A

Fig. 9 — The coding conventions.

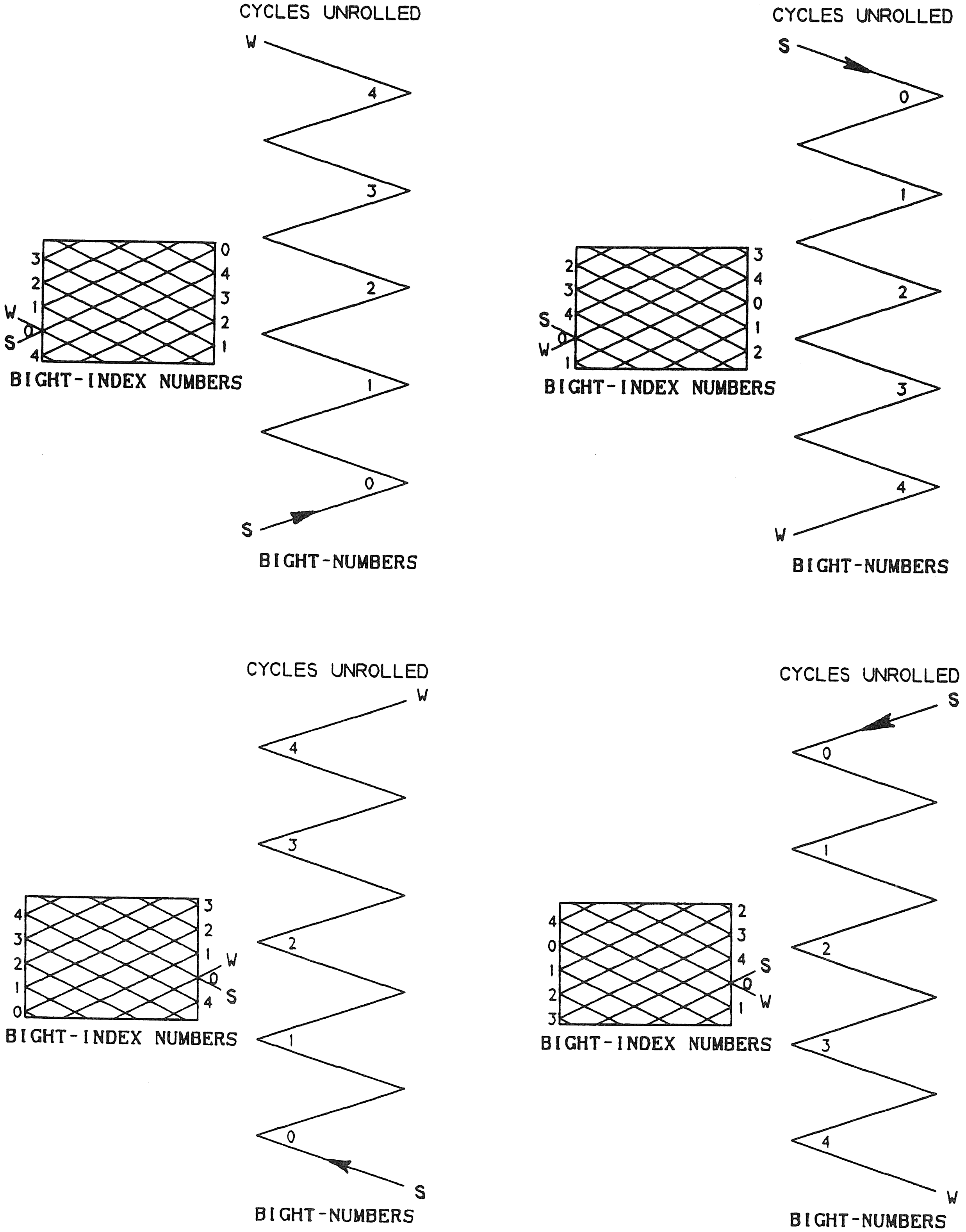


Fig. 10 — Bight-index numbers and Bight-numbers.

The bights, on each of the bight-boundaries, are numbered in consecutive order; these numbers are called **bight-index numbers**. The beginning of the string with which the Regular Knot is constructed is indicated by "**S**", the "Standing End". The opposite end of the string is called the "Working End", and is indicated by "**W**". The knot construction starts at bight-index number "0", and the bight on this first half-cycle, on the **opposite** bight-boundary, also carries the bight-index number "0". From these initial bights on either bight-boundary, the further bights receive bight-index numbers in consecutive order in the direction of braiding. This is illustrated for the four possible starts in Fig. 10.

The bights, on the bight-boundary **opposite** to the bight-boundary where the braiding of the knot commences, receive an additional set of numbers, called **bight-numbers**. These are associated with the braiding cycles. The bight between the first and second half-cycle (hence the apex of the first cycle) receives the bight-number 0; the bight between the third and fourth half-cycle (hence the apex of the second cycle) receives the bight-number 1; the bight between the fifth and sixth half-cycle (hence the apex of the third cycle) receives the bight-number 2; and so on. Fig. 10 illustrates this numbering scheme for the four possible starts.

A General Braiding Method

There are several methods for working out the necessary information which will enable the braider to construct the Column-coded Regular Knot he desires. Some braiders prefer to use sequence patterns of the braid; however it should be realised that they are then restricted in the parts and bights numbers which can be employed for the Column-coded Regular Knots. Restricting the number of bights to certain values necessitates that one has to cut the strings to the correct width when the knot has to fit a given diameter. The string-width depends largely on the number of bights of the knot and the diameter over which it has to fit, but the string-width is further influenced by the type of braiding pattern and the thickness of the strings. To complicate matters further, it also depends on the tightness with which the knot will be braided. It is therefore impossible to give an exact formula for the width of the string when the number of bights of the knot and the diameter over which it has to fit are given; only an approximate, emperical relationship can be given (see Ref. [1] pages 85, 86). Hence the reader will appreciate that, unless we braid always the same articles and always use the same type of knots, we **must** be able to vary the number of bights slightly in order to braid a knot to a high standard.

Besides the problem of cutting the string to the correct width, which is not known exactly, the braider might braid in a material of which the string-width cannot be chosen arbitrarily.

For the above reasons we strongly advise the reader against using such restrictive methods, of braiding to sequence patterns of braids. Nevertheless, for those who don't want to progress beyond such methods, and hence are satisfied with a very limited amount of braiding knowledge, we shall discuss later in this pamphlet some methods which can be used for "sequence pattern braiding".

There is not the slightest doubt that braiding a knot with the aid of half-cycle algorithms is the best and most general method. Furthermore we can apply two refinements to this method, which increase braiding efficiency considerably. We shall discuss these later in this pamphlet. Hence the reader is strongly advised to acquire the necessary skill in using the half-cycle algorithm method efficiently.

In this Section we shall discuss a method which will supply the braider with the necessary information for developing the half-cycle algorithm method used for braiding a Column-coded Regular Knot.

This information is obtained by making use of what is called *"the complementary cyclic bight-number scheme"* (see Ref. [2], pages 21-23). From this scheme we can read off the sequential column-numbers where a half-cycle "intersects" the previously laid down half-cycles. Since we are dealing with Column-coded Regular Knots, these column-numbers can be replaced by their respective codings. Consequently the sequence of sequential column-numbers obtained from the complementary bight-number scheme becomes the braiding algorithm for this half-cycle.

We shall explain the procedures with the aid of examples. Diagrams of these are shown in Figures 11, 12 and 13.

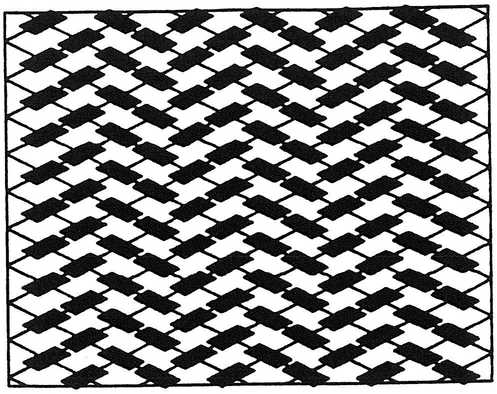


Fig. 11 — Regular Knot of Example 1.

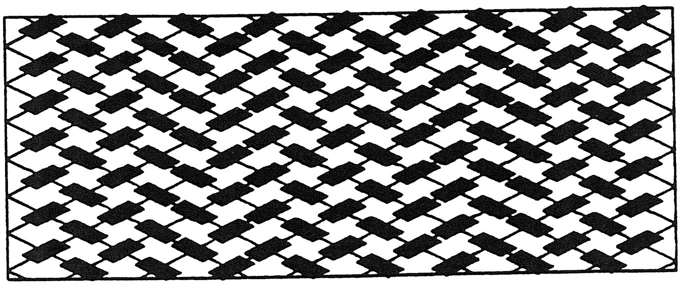


Fig. 12 — Regular Knot of Example 2.

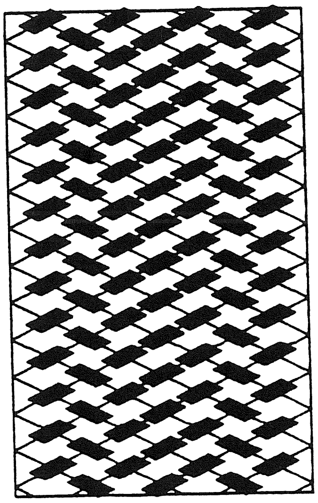


Fig. 13 — Regular Knot of Example 3.

Example 1 :

Suppose we want to braid a 2-pass Gaucho Knot with 13-parts and 10-bights ($p = 13$ and $b = 10$), and that its Column-coding is as given below:

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8	9	10	11	12	13
		/	/	\	\	/	/	\	\	/	/	\	\	
right to left ⟶	13	12	11	10	9	8	7	6	5	4	3	2	1	0

First we construct the complementary cyclic bight number-scheme. This can be done in two ways:

(A) Mark on a horizontal line a number of positions that is equal to the number of bights. Hence we have to mark off 10 positions as follows:

· · · · · · · · · ·

We shall allocate to these positions bight-numbers in a special order. In the construction of this order the value of $|-p|_b$ is used. Thus in this example we shall have to use the value $|-13|_{10} = 7$.

The first position on the left-hand side is always given the bight-number 0. Now count 7 further positions to the right, and give this last position the bight-number 1.

0							1		
·	·	·	·	·	·	·	·	·	·

Count again 7 further positions to the right, and give this last position the bight-number 2. In this counting of positions to the right, it should be realised that the bights on a bight-boundary lie equally spaced on a circle, hence after the last position on the right follows the first position on the left, which was given the bight-number 0. Hence the following result is obtained:

0				2			1		
·	·	·	·	·	·	·	·	·	·

Count again 7 further positions to the right, and give this last position the bight-number 3. Hence the following result is obtained:

0	3			2			1		
·	·	·	·	·	·	·	·	·	·

Keep repeating this construction process of the complementary cyclic bight-number scheme till all the positions have received a bight-number. This will result in the following complementary cyclic bight-number scheme for our example:

0	3	6	9	2	5	8	1	4	7
·	·	·	·	·	·	·	·	·	·

(B) The reader will observe that in the above derived complementary cyclic bight-number scheme the difference between any two consecutive bight-numbers is equal to 3. Note that the difference between the bight-numbers 9 and 2 is equal to 3 since $9+3 = 12$, but bight-number "12" is bight-number 2, due to the cyclic positioning of the bights on the bight-boundaries of the Regular Knot. Hence when the bight-number of a position in this complementary cyclic bight-number scheme is known (say its value is x), then its consecutive bight-number will be $|x+3|_b = |x+3|_{10}$. Thus, for example, when x is equal to 8, its consecutive bight-number is equal to $|8+3|_{10} = |11|_{10} = 1$.

This difference between two consecutive bight-numbers in the complementary cyclic bight-number scheme is indicated by the symbol $\Delta^* = |\overline{\Delta}|_b$ (see Ref. [2]).

The value of Δ^* can be calculated by means of the following formula:

$$\Delta^* = \left|\frac{m^* \cdot b + 1}{|-p|_b}\right|_b .$$

The value of m^* is to be that integer (positive, zero or negative) which is such that the value of

$$\left|\frac{m^* \cdot |b|_{|-p|_b} + 1}{|-p|_b}\right|$$

is the smallest whole number possible.

Integers are the numbers: $\cdots, -3, -2, -1, 0, +1, +2, +3, \cdots$
Whole numbers are the numbers: $0, +1, +2, +3, \cdots$

Hence in our example we obtain the following calculations:

$$\left|\frac{m^* \cdot |b|_{|-p|_b} + 1}{|-p|_b}\right| = \left|\frac{m^* \cdot |10|_{|-13|_{10}} + 1}{|-13|_{10}}\right| = \left|\frac{m^* \cdot |10|_7 + 1}{7}\right| = \left|\frac{m^* \cdot 3 + 1}{7}\right|$$

is the smallest whole number possible.

This will be the case when we take $m^* = 2$. Then:

$$\Delta^* = \left|\frac{m^* \cdot b + 1}{|-p|_b}\right|_b = \left|\frac{2 \cdot 10 + 1}{|-13|_{10}}\right|_{10} = \left|\frac{2 \cdot 10 + 1}{7}\right|_{10} = \left|\frac{21}{7}\right|_{10} = |3|_{10} = 3 .$$

We have already seen above how we can construct, with the aid of the Δ^*-value, the complementary cyclic bight-number scheme, starting at the left-most position which carries always the bight-number 0.

After obtaining, either by the method under **(A)** or by the method under **(B)**, the complementary cyclic bight-number scheme, we incorporate it in the diagram which displays the column-coding, and the column numbering from left to right and from right to left. For the convenience of the reader we repeat this diagram below:

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8	9	10	11	12	13
		/	/	\	\	/	/	\	\	/	/	\	\	
right to left ⟶	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Firstly, we insert the obtained complementary cyclic bight-number scheme between the "columns numbered left to right" line and the "coding" line, starting at column-number 0.

Secondly, we insert the obtained complementary cyclic bight-number scheme **in reverse order** between the "coding" line and the "columns numbered right to left" line, starting at column-number 0.

This will give the following result:

Columns: L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Bight-numbers ⟶	0	3	6	9	2	5	8	1	4	7				
		/	/	\	\	/	/	\	\	/	/	\	\	
Bight-numbers ⟵					7	4	1	8	5	2	9	6	3	0
Columns: L ⟵ R	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The reader will notice some empty spaces in the two "bight-numbers" lines, respectively above and below indicated codings. These spaces have still to be provided with bight-numbers. This is done by repeating the already established entries as far as is necessary. Hence the completed final diagram is as follows:

Columns : L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Bight-numbers ⟶	0	3	6	9	2	5	8	1	4	7	0	3	6	
		/	/	\	\	/	/	\	\	/	/	\	\	
Bight-numbers ⟵		6	3	0	7	4	1	8	5	2	9	6	3	0
Columns : L ⟵ R	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This diagram, called *the algorithm-diagram*, contains all the information required for braiding the required 13-part/10-bight 2-pass Gaucho Knot.

With a little experience we can braid the knot directly from it, or lacking such experience, we can first write down the half-cycle braiding algorithms from this algorithm-diagram, and braid the knot from that. Since this algorithm-diagram is a general algorithm-diagram for the construction of the knot in question, we can braid this knot in any way desired. We can braid it wholly "upwards", starting with the first half-cycle at a left bight and moving from lower-left to upper-right, or we can braid it wholly "downwards", starting with the first half-cycle at a left bight and moving from upper-left to lower-right; when left-handed, we can braid it wholly "upwards", starting with the first half-cycle at a right bight and moving from lower-right to upper-left, or we can braid it wholly "downwards", starting with the first half-cycle at a right bight and moving from upper-right to lower-left. We can also braid it partly "upwards" and partly "downwards", a preferred way when the knot requires a long length of string.

Before deriving a half-cycle braiding algorithm from the obtained algorithm-diagram, we shall first discuss an important relationship between half-cycles and bight-numbers. The half-cycles are numbered in sequential order as they are laid down. Thus, first we lay down half-cycle 1, then half-cycle 2, then half-cycle 3, then half-cycle 4, and so on. Hence we have **odd half-cycles** and **even half-cycles**. Half-cycle 1 is a special one; since it is the very first half-cycle in the braiding process, it is always a "free run"; that is, it cannot "intersect" a previously laid down half-cycle since there isn't one. Hence in the derivation of the half-cycle braiding algorithms, we start with the second half-cycle.

With each bight-number there are, in general, associated two consecutive half-cycles; an even one and an odd one. The odd half-cycle "intersects" always the same column-**numbers** as the even half-cycle (note that the column-**numbers** are the same, **not** the columns, since for one half-cycle the columns are numbered from left to right, and for the other half-cycle the columns are numbered from right to left).

Suppose an even half-cycle is h_e , then its associated bight-number i is equal to:

$$i = \frac{h_e - 2}{2} \; ;$$

and suppose an odd half-cycle is h_o , then its associated bight-number i is equal to:

$$i = \frac{h_o - 3}{2} \; .$$

A half-cycle (even or odd) which is associated with bight-number i intersects all column-numbers which in the algorithm-diagram correspond to the bight-numbers which are less than or equal to i .

Thus for example, the 12th half-cycle is associated with bight-number:

$$i = \frac{h_e - 2}{2} = \frac{12-2}{2} = \frac{10}{2} = 5\,.$$

If we braid this half-cycle **upwards** from **right** to **left**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$o - u - o - 2u - o - u\,.$$

Note that we intersect the Column-numbers (right to left) 1, 4, 5, 7, 8, 10, 11 .

If, for example, we braid this half-cycle **downwards** from **left** to **right**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$u - o - u - 2o - u - o\,.$$

Note that we intersect the Column-numbers (left to right) 1, 4, 5, 7, 8, 10, 11 .

The 17th half-cycle, for example, is associated with bight-number:

$$i = \frac{h_o - 3}{2} = \frac{17-3}{2} = \frac{14}{2} = 7\,.$$

If we braid this half-cycle **upwards** from **left** to **right**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$2o - u - o - 2u - 2o - 2u\,.$$

Note that we intersect the Column-numbers (left to right) 1, 2, 4, 5, 7, 8, 9, 10, 11, 12 .

If, for example, we braid this half-cycle **downwards** from **right** to **left**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$2u - o - u - 2o - 2u - 2o\,.$$

Note that we intersect the Column-numbers (right to left) 1, 2, 4, 5, 7, 8, 9, 10, 11, 12 .

Below we present the complete algorithm-table for braiding this knot **wholly** upwards; for each half-cycle its associated bight-number is indicated.

1.		L ⟶ R :	free run .
2.	$(i=0)$	R ⟶ L :	$o\,.$
3.	$(i=0)$	L ⟶ R :	$o\,.$
4.	$(i=1)$	R ⟶ L :	$u - o\,.$
5.	$(i=1)$	L ⟶ R :	$u - o\,.$
6.	$(i=2)$	R ⟶ L :	$2u - o\,.$
7.	$(i=2)$	L ⟶ R :	$2u - o\,.$
8.	$(i=3)$	R ⟶ L :	$o - 2u - o - u\,.$
9.	$(i=3)$	L ⟶ R :	$o - 2u - o - u\,.$
10.	$(i=4)$	R ⟶ L :	$o - 3u - o - u\,.$
11.	$(i=4)$	L ⟶ R :	$o - 3u - o - u\,.$
12.	$(i=5)$	R ⟶ L :	$o - u - o - 2u - o - u\,.$
13.	$(i=5)$	L ⟶ R :	$o - u - o - 2u - o - u\,.$
14.	$(i=6)$	R ⟶ L :	$2o - u - o - 2u - o - 2u\,.$
15.	$(i=6)$	L ⟶ R :	$2o - u - o - 2u - o - 2u\,.$
16.	$(i=7)$	R ⟶ L :	$2o - u - o - 2u - 2o - 2u\,.$
17.	$(i=7)$	L ⟶ R :	$2o - u - o - 2u - 2o - 2u\,.$
18.	$(i=8)$	R ⟶ L :	$2o - u - 2o - 2u - 2o - 2u\,.$
19.	$(i=8)$	L ⟶ R :	$2o - u - 2o - 2u - 2o - 2u\,.$
20.	$(i=9)$	R ⟶ L :	$2o - 2u - 2o - 2u - 2o - 2u\,.$

Example 2 :

Suppose we want to braid an 18-parts and 7-bights ($p = 18$ and $b = 7$) Column-coded Regular Knot, whose Column-coding is as given below:

columns numbered

left to right ⟶	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		/	\	/	\	\	/	/	\	/	\	/	/	\	\	/	\	/	
right to left ⟶	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

First we construct the complementary cyclic bight number-scheme which, as we have seen in Example 1, can be done in two ways:

(A) Mark on a horizontal line a number of positions that is equal to the number of bights. Hence we have to mark off 7 positions as follows:

. . . .

To these positions we shall allocate bight-numbers. The value of $|-p|_b$ is used in this process. Thus in this example we have to use the value $|-18|_7 = 3$.

The first position on the left-hand side is, as we have seen in Example 1, always given the bight-number 0. Now count 3 further positions to the right, and give this last position the bight-number 1.

0			1			
.	.	.	.	.	.	.

Count again 3 further positions to the right, and give this last position the bight-number 2. Hence the following result is obtained:

0			1			2
.	.	.	.	.	.	.

Count again 3 further positions to the right, and give this last position the bight-number 3. In this counting of positions to the right, it should be realised that the bights on a bight-boundary lie equally spaced on a circle, hence after the last position on the right follows the first position on the left, which was given the bight-number 0. Hence the following result is obtained:

0		3	1			2
.	.	.	.	.	.	.

Keep repeating this construction process of the complementary cyclic bight-number scheme till all the positions have received a bight-number. This will result in the following complementary cyclic bight-number scheme for our example:

0	5	3	1	6	4	2
.	.	.	.	.	.	.

(B) The reader will observe that in the above derived complementary cyclic bight-number scheme the difference between any two consecutive bight-numbers is equal to 5. Note that the difference between the bight-numbers 5 and 3 is equal to 5 since $5+5 = 10$, but bight-number "10" is bight-number 3, due to the cyclic positioning of the bights on the bight-boundaries of the Regular Knot. Hence when the bight-number of a position in this complementary cyclic bight-number scheme is known (say its value is x), then its consecutive bight-number will be $|x + 5|_b = |x + 5|_7$. Thus, for example, when x is equal to 6, its consecutive bight-number is equal to $|6 + 5|_7 = |11|_7 = 4$.

As mentioned in Example 1, this difference between two consecutive bight-numbers in the complementary cyclic bight-number scheme is indicated by the symbol Δ^*.

The value of Δ^* can be calculated by means of the following formula:

$$\Delta^* = \left|\frac{m^* \cdot b + 1}{|-p|_b}\right|_b .$$

The value of m^* is to be that integer (positive, zero or negative) which is such that the value of

$$\left|\frac{m^* \cdot |b|_{|-p|_b} + 1}{|-p|_b}\right|$$

is the smallest whole number possible.

Integers are the numbers: $\cdots, -3, -2, -1, 0, +1, +2, +3, \cdots$
Whole numbers are the numbers: $0, +1, +2, +3, \cdots$

Hence in our example we obtain the following calculations:

$$\left|\frac{m^* \cdot |b|_{|-p|_b} + 1}{|-p|_b}\right| = \left|\frac{m^* \cdot |7|_{|-18|_7} + 1}{|-18|_7}\right| = \left|\frac{m^* \cdot |7|_3 + 1}{3}\right| = \left|\frac{m^* \cdot 1 + 1}{3}\right|$$

is the smallest whole number possible.
This will be the case when we take $m^* = -1$. Then:

$$\Delta^* = \left|\frac{m^* \cdot b + 1}{|-p|_b}\right|_b = \left|\frac{-1 \cdot 7 + 1}{|-18|_7}\right|_7 = \left|\frac{-1 \cdot 7 + 1}{3}\right|_7 = \left|\frac{-6}{3}\right|_7 = |-2|_7 = 5 .$$

We have already seen above how we can construct, with the aid of the Δ^*-value, the complementary cyclic bight-number scheme, starting at the left-most position which carries always the bight-number 0.

After obtaining, either by the method under **(A)** or by the method under **(B)**, the complementary cyclic bight-number scheme, we incorporate it in the diagram which displays the column-coding, and the column numbering from left to right and from right to left. For the convenience of the reader we repeat this diagram below:

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		/	\	/	\	\	/	/	\	/	\	/	/	\	\	/	\	/	
right to left ⟶	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Firstly, we insert the obtained complementary cyclic bight-number scheme between the "columns numbered left to right" line and the "coding" line, starting at column-number 0.

Secondly, we insert the obtained complementary cyclic bight-number scheme **in reverse order** between the "coding" line and the "columns numbered right to left" line, starting at column-number 0.

This will give the following result:

Columns: L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Bight-numbers ⟶	0	5	3	1	6	4	2												
		/	\	/	\	\	/	/	\	/	\	/	/	\	\	/	\	/	
Bight-numbers ⟵													2	4	6	1	3	5	0
Columns: L ⟵ R	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The reader will notice some empty spaces in the two "bight-numbers" lines, respectively above and below indicated codings. These spaces have still to be provided with bight-numbers. This is done by repeating the the already established entries as far as is necessary. Hence the completed final diagram is as follows:

Columns: L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Bight-numbers ⟶	0	5	3	1	6	4	2	0	5	3	1	6	4	2	0	5	3	1	
		/	\	/	\	\	/	/	\	/	\	/	/	\	\	/	\	/	
Bight-numbers ⟵		1	3	5	0	2	4	6	1	3	5	0	2	4	6	1	3	5	0
Columns: L ⟵ R	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This *algorithm-diagram* contains all the information required for braiding the required 18-part/7-bight knot.

As mentioned before, with a little experience we can braid the knot directly from it, or lacking such experience, we can first write down the half-cycle braiding algorithms from this algorithm-diagram, and braid the knot from that. Since this algorithm-diagram is a general algorithm-diagram for the construction of the knot in question, we can braid this knot in any way desired. We can braid it wholly "upwards", starting with the first half-cycle at a left bight and moving from lower-left to upper-right, or we can braid it wholly "downwards", starting with the first half-cycle at a left bight and moving from upper-left to lower-right; when left-handed, we can braid it wholly "upwards", starting with the first half-cycle at a right bight and moving from lower-right to upper-left, or we can braid it wholly "downwards", starting with the first half-cycle at a right bight and moving from upper-right to lower-left. We can also braid it partly "upwards" and partly "downwards", a preferred way when the dimensions of a knot are such that a long length of string is required in its construction.

As mentioned under example 1, each bight-number is, in general, associated with two consecutive half-cycles; an even one and an odd one. The odd half-cycle "intersects" the same column-**numbers** as the even half-cycle (note that the column-**numbers** are the same, **not** the columns, since for one half-cycle the columns are numbered from left to right, and for the other, the columns are numbered from right to left).

When an even half-cycle is indicated by h_e, then its associated bight-number i is equal to:

$$i = \frac{h_e - 2}{2} ,$$

and when an odd half-cycle is indicated by h_o, then its associated bight-number i is equal to:

$$i = \frac{h_o - 3}{2} .$$

A half-cycle (even or odd) which is associated with bight-number i intersects all column-numbers which in the algorithm-diagram correspond to the bight-numbers which are less than or equal to i.

Thus for example, the 10^{th} half-cycle is associated with bight-number:

$$i = \frac{h_e - 2}{2} = \frac{10 - 2}{2} = \frac{8}{2} = 4 .$$

If we braid this half-cycle **upwards** from **right** to **left**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$o - u - o - 3u - o - u - 3o - u .$$

Note: intersected Column-numbers are (right to left) 2, 3, 5, 6, 7, 9, 10, 12, 13, 14, 16, 17.

If, for example, we braid this half-cycle **downwards** from **left** to **right**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$o - u - o - 3u - o - u - 3o - u\,.$$

Note: intersected Column-numbers are (left to right) 2, 3, 5, 6, 7, 9, 10, 12, 13, 14, 16, 17.

The 7th half-cycle, for example, is associated with bight-number:

$$i = \frac{h_o - 3}{2} = \frac{7-3}{2} = \frac{4}{2} = 2\,.$$

If we braid this half-cycle **upwards from left to right**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$3o - 3u - o\,.$$

Note that we intersect the Column-numbers (left to right) 3, 6, 7, 10, 13, 14, 17.

If, for example, we braid this half-cycle **downwards** from **right to left**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$3o - 3u - o\,.$$

Note that we intersect the Column-numbers (right to left) 3, 6, 7, 10, 13, 14, 17.

Below we present the complete algorithm-table for braiding this knot wholly upwards; for each half-cycle its associated bight-number is indicated.

1. L ⟶ R : free run.
2. $(i = 0)$ R ⟶ L : $u - o$.
3. $(i = 0)$ L ⟶ R : $o - u$.
4. $(i = 1)$ R ⟶ L : $2u - 2o - u$.
5. $(i = 1)$ L ⟶ R : $2o - 2u - o$.
6. $(i = 2)$ R ⟶ L : $3u - 3o - u$.
7. $(i = 2)$ L ⟶ R : $3o - 3u - o$.
8. $(i = 3)$ R ⟶ L : $o - 4u - 4o - u$.
9. $(i = 3)$ L ⟶ R : $u - 4o - 4u - o$.
10. $(i = 4)$ R ⟶ L : $o - u - o - 3u - o - u - 3o - u$.
11. $(i = 4)$ L ⟶ R : $u - o - u - 3o - u - o - 3u - o$.
12. $(i = 5)$ R ⟶ L : $u - o - u - o - 2u - o - u - o - u - 2o - u - o - u$.
13. $(i = 5)$ L ⟶ R : $o - u - o - u - 2o - u - o - u - o - 2u - o - u - o$.
14. $(i = 6)$ R ⟶ L : $u - o - u - 2o - 2u - o - u - o - 2u - 2o - u - o - u$.

Example 3:

Suppose we want to braid an 8-parts and 13-bights ($p = 8$ and $b = 13$) Column-coded Regular Knot, whose Column-coding is as given below:

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8
		/	\	/	/	/	\	/	
right to left ⟶	8	7	6	5	4	3	2	1	0

First we construct the complementary cyclic bight number-scheme which, as we have seen in Example 1, can be done in two ways:

(A) Mark on a horizontal line a number of positions that is equal to the number of bights. Hence we have to mark off 13 positions as follows:

. . . .

To these positions we shall allocate bight-numbers. The value of $|-p|_b$ is used in this process. Thus in this example we have to use the value $|-8|_{13} = 5$.

The first position on the left-hand side is, as we have seen in Example 1, always given the bight-number 0. Now count 5 further positions to the right, and give this last position the bight-number 1.

0					1							
.	.	.	.	.	.	.	.	.	.	.	.	.

Count again 5 further positions to the right, and give this last position the bight-number 2. Hence the following result is obtained:

0					1					2		
.	.	.	.	.	.	.	.	.	.	.	.	.

Count again 5 further positions to the right, and give this last position the bight-number 3. In this counting of positions to the right, it should be realised that the bights on a bight-boundary lie equally spaced on a circle, hence after the last position on the right follows the first position on the left, which was given the bight-number 0. Hence the following result is obtained:

0		3			1					2		
.	.	.	.	.	.	.	.	.	.	.	.	.

Keep repeating this construction process of the complementary cyclic bight-number scheme till all the positions have received a bight-number. This will result in the following complementary cyclic bight-number scheme for our example:

0	8	3	11	6	1	9	4	12	7	2	10	5
.	.	.	.	.	.	.	.	.	.	.	.	.

(B) The reader will observe that in the above derived complementary cyclic bight-number scheme the difference between any two consecutive bight-numbers is equal to 8. Note that the difference between the bight-numbers 8 and 3 is equal to 8 since $8+8=16$, but bight-number "16" is bight-number 3, due to the cyclic positioning of the bights on the bight-boundaries of the Regular Knot. Hence when the bight-number of a position in this complementary cyclic bight-number scheme is known (say its value is x), then its consecutive bight-number will be $|x+8|_b = |x+8|_{13}$. Thus, for example, when x is equal to 11, its consecutive bight-number is equal to $|11+8|_{13} = |19|_{13} = 6$.

As mentioned in Example 1, this difference between two consecutive bight-numbers in the complementary cyclic bight-number scheme is indicated by the symbol Δ^*.

The value of Δ^* can be calculated by means of the following formula:

$$\Delta^* = \left| \frac{m^* \cdot b + 1}{|-p|_b} \right|_b .$$

The value of m^* is to be that integer (positive, zero or negative) which is such that the value of

$$\left| \frac{m^* \cdot |b|_{|-p|_b} + 1}{|-p|_b} \right|$$

is the smallest whole number possible.

Integers are the numbers: $\cdots, -3, -2, -1, 0, +1, +2, +3, \cdots$
Whole numbers are the numbers: $0, +1, +2, +3, \cdots$

Hence in our example we obtain the following calculations:

$$\left|\frac{m^*\cdot|b|_{|-p|_b}+1}{|-p|_b}\right| = \left|\frac{m^*\cdot|13|_{|-8|_{13}}+1}{|-8|_{13}}\right| = \left|\frac{m^*\cdot|13|_5+1}{5}\right| = \left|\frac{m^*\cdot 3+1}{5}\right|$$

is the smallest whole number possible.

This will be the case when we take $m^* = -2$. Then:

$$\Delta^* = \left|\frac{m^*\cdot b+1}{|-p|_b}\right|_b = \left|\frac{-2\cdot 13+1}{|-8|_{13}}\right|_{13} = \left|\frac{-2\cdot 13+1}{5}\right|_{13} = \left|\frac{-25}{5}\right|_{13} = |-5|_{13} = 8\,.$$

We have already seen above how we can construct, with the aid of the Δ^*-value, the complementary cyclic bight-number scheme, starting at the left-most position which carries always the bight-number 0.

After obtaining, either by the method under **(A)** or by the method under **(B)**, the complementary cyclic bight-number scheme, we incorporate it in the diagram which displays the column-coding, and the column numbering from left to right and from right to left. For the convenience of the reader we repeat this diagram below:

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8
		/	\	/	/	/	\	/	
right to left ⟶	8	7	6	5	4	3	2	1	0

Firstly, we insert the obtained complementary cyclic bight-number scheme between the "columns numbered left to right" line and the "coding" line, starting at column-number 0.

Secondly, we insert the obtained complementary cyclic bight-number scheme **in reverse order** between the "coding" line and the "columns numbered right to left" line, starting at column-number 0.

Since the "bight-numbers" lines have to extend to or have to be repeated to the position with the last coding, we only use a portion of the "bight-numbers" lines in this Example.

Hence we shall obtain the following result:

Columns: L ⟶ R	0	1	2	3	4	5	6	7	8
Bight-numbers ⟶	0	8	3	11	6	1	9	4	
		/	\	/	/	/	\	/	
Bight-numbers ⟵		4	9	1	6	11	3	8	0
Columns: L ⟵ R	8	7	6	5	4	3	2	1	0

This *algorithm-diagram* contains all the information required for braiding the required 8-part/13-bight knot.

With a little experience we can braid the knot directly from it, or lacking such experience, we can first write down the half-cycle braiding algorithms from this algorithm-diagram, and braid the knot from that. Since this algorithm-diagram is a general algorithm-diagram for the construction of the knot in question, we can braid this knot in any way desired. We can braid it wholly "upwards", starting with the first half-cycle at a left bight and moving from lower-left to upper-right, or we can braid it wholly "downwards", starting with the first half-cycle at a left bight and moving from upper-left to lower-right; when left-handed, we can braid it wholly "upwards", starting with the first half-cycle at a right bight and moving from lower-right to upper-left, or we can braid it wholly "downwards", starting with the first half-cycle at a right bight and moving from upper-right to lower-left. We can also braid it partly "upwards" and partly "downwards", a preferred way when the dimensions of a knot are such that a long length of string is required in its construction.

As mentioned under example 1, each bight-number is, in general, associated with two consecutive half-cycles; an even one and an odd one. The odd half-cycle "intersects" the same column-**numbers** as the even half-cycle (note that the column-**numbers** are the same, **not** the columns, since for one half-cycle the columns are numbered from left to right, and for the other, the columns are numbered from right to left).

When an even half-cycle is indicated by h_e, then its associated bight-number i is equal to:

$$i = \frac{h_e - 2}{2},$$

and when an odd half-cycle is indicated by h_o, then its associated bight-number i is equal to:

$$i = \frac{h_o - 3}{2}.$$

A half-cycle (even or odd) which is associated with bight-number i intersects all column-numbers which in the algorithm-diagram correspond to the bight-numbers which are less than or equal to i.

Thus for example, the 16th half-cycle is associated with bight-number:

$$i = \frac{h_e - 2}{2} = \frac{16 - 2}{2} = \frac{14}{2} = 7.$$

If we braid this half-cycle **upwards** from **right** to **left**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$o - 3u.$$

Note: intersected Column-numbers are (right to left) 2, 4, 5, 7.

If, for example, we braid this half-cycle **downwards** from **left** to **right**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$o - 3u.$$

Note: intersected Column-numbers are (left to right) 2, 4, 5, 7.

The 9th half-cycle, for example, is associated with bight-number:

$$i = \frac{h_o - 3}{2} = \frac{9 - 3}{2} = \frac{6}{2} = 3.$$

If we braid this half-cycle **upwards** from **left** to **right**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$u - o.$$

Note that we intersect the Column-numbers (left to right) 2, 5.

If, for example, we braid this half-cycle **downwards** from **right** to **left**, we obtain from the algorithm-diagram for this half-cycle the following algorithm:

$$u - o.$$

Note that we intersect the Column-numbers (right to left) 2, 5.

Below we present the complete algorithm-table for braiding this knot wholly upwards; for each half-cycle its associated bight-number is indicated.

1. $\mathrm{L} \longrightarrow \mathrm{R}$: free run .
2. $(i=0)$ $\mathrm{R} \longrightarrow \mathrm{L}$: free run .
3. $(i=0)$ $\mathrm{L} \longrightarrow \mathrm{R}$: free run .
4. $(i=1)$ $\mathrm{R} \longrightarrow \mathrm{L}$: u .
5. $(i=1)$ $\mathrm{L} \longrightarrow \mathrm{R}$: o .
6. $(i=2)$ $\mathrm{R} \longrightarrow \mathrm{L}$: u .
7. $(i=2)$ $\mathrm{L} \longrightarrow \mathrm{R}$: o .
8. $(i=3)$ $\mathrm{R} \longrightarrow \mathrm{L}$: $o-u$.
9. $(i=3)$ $\mathrm{L} \longrightarrow \mathrm{R}$: $u-o$.
10. $(i=4)$ $\mathrm{R} \longrightarrow \mathrm{L}$: $o-2u$.
11. $(i=4)$ $\mathrm{L} \longrightarrow \mathrm{R}$: $u-2o$.
12. $(i=5)$ $\mathrm{R} \longrightarrow \mathrm{L}$: $o-2u$.
13. $(i=5)$ $\mathrm{L} \longrightarrow \mathrm{R}$: $u-2o$.
14. $(i=6)$ $\mathrm{R} \longrightarrow \mathrm{L}$: $o-3u$.
15. $(i=6)$ $\mathrm{L} \longrightarrow \mathrm{R}$: $u-3o$.
16. $(i=7)$ $\mathrm{R} \longrightarrow \mathrm{L}$: $o-3u$.
17. $(i=7)$ $\mathrm{L} \longrightarrow \mathrm{R}$: $u-3o$.
18. $(i=8)$ $\mathrm{R} \longrightarrow \mathrm{L}$: $u-o-3u$.
19. $(i=8)$ $\mathrm{L} \longrightarrow \mathrm{R}$: $o-u-3o$.
20. $(i=9)$ $\mathrm{R} \longrightarrow \mathrm{L}$: $u-o-2u-o-u$.
21. $(i=9)$ $\mathrm{L} \longrightarrow \mathrm{R}$: $o-u-2o-u-o$.
22. $(i=10)$ $\mathrm{R} \longrightarrow \mathrm{L}$: $u-o-2u-o-u$.
23. $(i=10)$ $\mathrm{L} \longrightarrow \mathrm{R}$: $o-u-2o-u-o$.
24. $(i=11)$ $\mathrm{R} \longrightarrow \mathrm{L}$: $u-o-3u-o-u$.
25. $(i=11)$ $\mathrm{L} \longrightarrow \mathrm{R}$: $o-u-3o-u-o$.
26. $(i=12)$ $\mathrm{R} \longrightarrow \mathrm{L}$: $u-o-3u-o-u$.

Refinements

On page 12 of the previous Section we stated that two refinements could be applied to the use of the half-cycle algorithm method for braiding a Column-coded Regular Knot. These refinements are:

> **(I)** By starting the braiding at the midpoint of the string-length which is used in the construction of a Column-coded Regular Knot.
> This method of braiding has obviously great advantages when the string-length required is long. It avoids a long length of string to be continuously threaded through the braid-work completed so far, thus preventing, or at least minimizing, damage to both. Furthermore, it speeds up the braiding process considerably.
>
> **(II)** By checking the progression of the construction of a Column-coded Regular Knot with the aid of the "Standing End" of the string. This aids the avoidance of mistakes during the construction stage of the knot.

The required information for both these refinements is already contained in the algorithm diagram of the Column-coded Regular Knot. This once again shows the great versatility of those diagrams, giving further reason why the serious braider should become well aquainted with them. We shall now discuss these refinements.

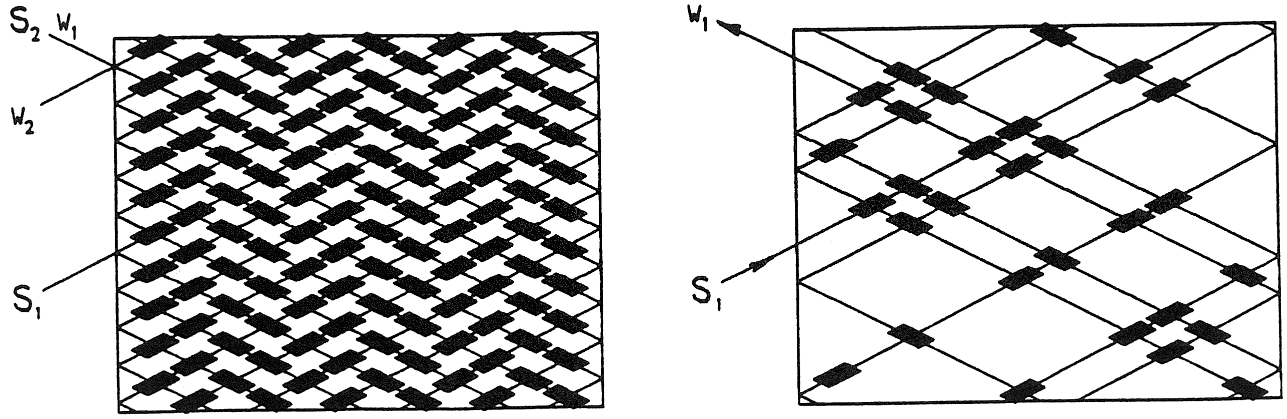
(I) By starting the braiding of the knot at the midpoint of the string, half the knot is braided "upwards" and half the knot is braided "downwards". Suppose we start with braiding upwards, then the last half-cycle in this upwards braiding-phase should be an *even* half-cycle. For example, a knot with 10 bights has 20 half-cycles in total, and hence the last half-cycle in this, upwards braided, first part of the construction process should be the 10th half-cycle. From the 11th half-cycle onwards (starting with the other half of the string, hence the "Standing End" in the first half of the knot construction becomes the "Working End" in the second half of the knot construction) we braid downwards. Hence in the algorithm-table for braiding **wholly upwards**, we have only to change all the crossing-types in the half-cycle algorithms for half-cycles 11 and higher to the opposite type, replacing each u by o and vice versa.

Note that the "Working End" of the end of the last half-cycle in the first half of the braiding process becomes the "Standing End" in the second half of the braiding process (see Fig. 14).

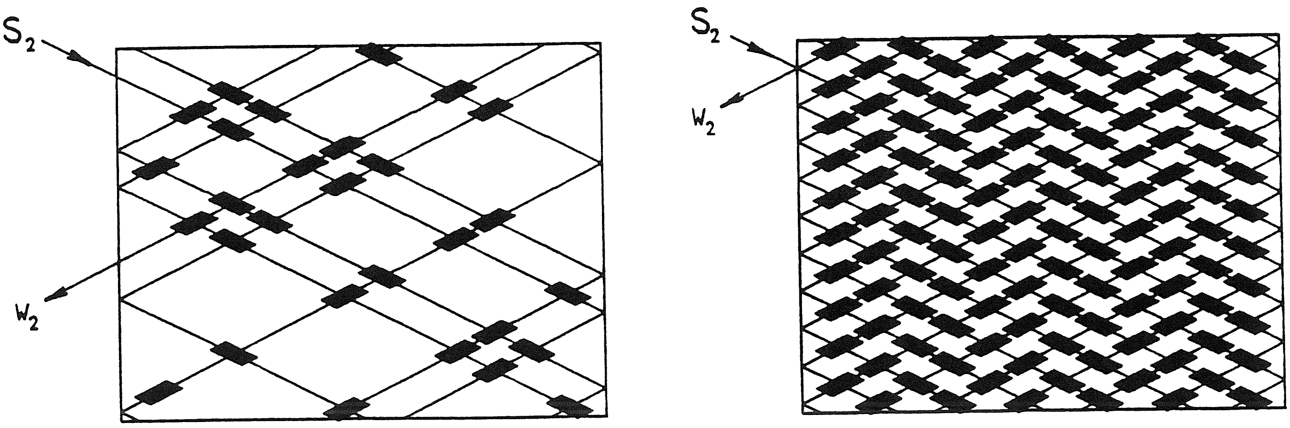
Instead of obtaining the algorithm-table from the one for wholly braiding upwards, we can also obtain it directly from the algorithm-diagram of the knot.

If this 10-bight Column-coded Regular Knot is the one of Example 1, then, as shown there, its algorithm-diagram is:

Columns: L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Bight-numbers ⟶	0	3	6	9	2	5	8	1	4	7	0	3	6	
		/	/	\	\	/	/	\	\	/	/	\	\	
Bight-numbers ⟵		6	3	0	7	4	1	8	5	2	9	6	3	0
Columns: L ⟵ R	13	12	11	10	9	8	7	6	5	4	3	2	1	0



End of first braiding phase



Start of second braiding phase

End of second braiding phase

Fig. 14 — The two braiding phases.

The algorithm-table for braiding this knot in the manner described above is as follows:

	1.		$L \longrightarrow R$:	free run .
	2.	$(i=0)$	$R \longrightarrow L$:	o .
upwards	3.	$(i=0)$	$L \longrightarrow R$:	o .
	4.	$(i=1)$	$R \longrightarrow L$:	$u-o$.
	5.	$(i=1)$	$L \longrightarrow R$:	$u-o$.
	6.	$(i=2)$	$R \longrightarrow L$:	$2u-o$.
	7.	$(i=2)$	$L \longrightarrow R$:	$2u-o$.
	8.	$(i=3)$	$R \longrightarrow L$:	$o-2u-o-u$.
	9.	$(i=3)$	$L \longrightarrow R$:	$o-2u-o-u$.
	10.	$(i=4)$	$R \longrightarrow L$:	$o-3u-o-u$.
	11.	$(i=4)$	$L \longrightarrow R$:	$u-3o-u-o$.
	12.	$(i=5)$	$R \longrightarrow L$:	$u-o-u-2o-u-o$.
downwards	13.	$(i=5)$	$L \longrightarrow R$:	$u-o-u-2o-u-o$.
	14.	$(i=6)$	$R \longrightarrow L$:	$2u-o-u-2o-u-2o$.
	15.	$(i=6)$	$L \longrightarrow R$:	$2u-o-u-2o-u-2o$.
	16.	$(i=7)$	$R \longrightarrow L$:	$2u-o-u-2o-2u-2o$.
	17.	$(i=7)$	$L \longrightarrow R$:	$2u-o-u-2o-2u-2o$.
	18.	$(i=8)$	$R \longrightarrow L$:	$2u-o-2u-2o-2u-2o$.
	19.	$(i=8)$	$L \longrightarrow R$:	$2u-o-2u-2o-2u-2o$.
	20.	$(i=9)$	$R \longrightarrow L$:	$2u-2o-2u-2o-2u-2o$.

When the Column-coded Regular Knot has an *odd* number of bights, say for example 13, then we could braid, say upwards,the first 12 or 14 half-cycles (the last half-cycle braided upwards should be an *even* half-cycle). We take 14 half-cycles, since the number of crossings laid down during the first part of the knot construction process is less than the number of crossings laid down during the second part of the braiding process. Thus from the 15th half-cycle onwards (starting with the other half of the string, hence the "Standing End" in the first part of the knot construction becomes the "Working End" in the second part of the knot construction) we braid downwards. Consequently in the algorithm table for braiding **wholly upwards**, we have only to change all the crossing-types in the half-cycle algorithms for half-cycles 15 and higher to the opposite type, replacing each u by o and vice versa.

Note again that the "Working End" of the end of the last half-cycle in the first half of the braiding process becomes the "Standing End" in the second half of the braiding process.

Instead of obtaining the algorithm-table from the one for wholly braiding upwards, we can also obtain it directly from the algorithm-diagram of the knot.

If this 13-bight Column-coded Regular Knot is the one of Example 3, then, as shown there, its algorithm-diagram is:

Columns: $L \longrightarrow R$	0	1	2	3	4	5	6	7	8
Bight-numbers $\longrightarrow$	0	8	3	11	6	1	9	4	
		/	\	/	/	/	\	/	
Bight-numbers $\longleftarrow$		4	9	1	6	11	3	8	0
Columns: $L \longleftarrow R$	8	7	6	5	4	3	2	1	0

The algorithm-table for braiding this knot in the manner described above is as follows:

upwards	1.		L ⟶ R :	free run .
	2.	$(i=0)$	R ⟶ L :	free run .
	3.	$(i=0)$	L ⟶ R :	free run .
	4.	$(i=1)$	R ⟶ L :	u .
	5.	$(i=1)$	L ⟶ R :	o .
	6.	$(i=2)$	R ⟶ L :	u .
	7.	$(i=2)$	L ⟶ R :	o .
	8.	$(i=3)$	R ⟶ L :	$o-u$.
	9.	$(i=3)$	L ⟶ R :	$u-o$.
	10.	$(i=4)$	R ⟶ L :	$o-2u$.
	11.	$(i=4)$	L ⟶ R :	$u-2o$.
	12.	$(i=5)$	R ⟶ L :	$o-2u$.
	13.	$(i=5)$	L ⟶ R :	$u-2o$.
	14.	$(i=6)$	R ⟶ L :	$o-3u$.
downwards	15.	$(i=6)$	L ⟶ R :	$o-3u$.
	16.	$(i=7)$	R ⟶ L :	$u-3o$.
	17.	$(i=7)$	L ⟶ R :	$o-3u$.
	18.	$(i=8)$	R ⟶ L :	$o-u-3o$.
	19.	$(i=8)$	L ⟶ R :	$u-o-3u$.
	20.	$(i=9)$	R ⟶ L :	$o-u-2o-u-o$.
	21.	$(i=9)$	L ⟶ R :	$u-o-2u-o-u$.
	22.	$(i=10)$	R ⟶ L :	$o-u-2o-u-o$.
	23.	$(i=10)$	L ⟶ R :	$u-o-2u-o-u$.
	24.	$(i=11)$	R ⟶ L :	$o-u-3o-u-o$.
	25.	$(i=11)$	L ⟶ R :	$u-o-3u-o-u$.
	26.	$(i=12)$	R ⟶ L :	$o-u-3o-u-o$.

(II) If unknowingly an accidental mistake is made in the execution of one or more half-cycle braiding algorithms, not only a lot of time may be wasted, but worse, the string has been subjected to a lot of unnecessary "wear". Therefore having a means of keeping a constant check on the braiding progress can be of great value to the artisan. Such a means is readily available when the Column-coded Regular Knot is braided by the "half-cycle algorithm" method. It is already incorporated in the *algorithm-diagram* of the knot. It consists of checking the **even** half-cycles during the braiding process with respect to the "Standing End". This is done by indicating in each even half-cycle algorithm which crossing or crossings it makes with the "Standing End" (depending on the values for p and b it may well be that some even half-cycles don't cross the "Standing End"). We shall indicate such a crossing or crossings by means of the following convention:

An entry $\cdots - (1,s,2)4o - \cdots$ indicates that in this set of 4 *overs*, the 2nd *over* crosses the "Standing End".

An entry $\cdots - (s)u - \cdots$ indicates that this single *under* crosses the "Standing End".

An entry $\cdots - (s,2,s,3,s,2)10u - \cdots$ indicates that the string crosses the "Standing End" on the 1st, 4th and 8th *under* in this set of 10 *unders*.

In the *algorithm-diagram* of the Column-coded Regular Knot, the crossing(s)s which an *even* half-cycle belonging to bight-number i makes with the "Standing End" is(are) on the column-number(s) which correspond(s) with the bight-number i.

Example 4:

Suppose the Column-coded Regular Knot is as in Example 1, and we braid it wholly by the upwards braiding method. For convenience we repeat here its algorithm-diagram:

Columns: L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Bight-numbers ⟶	0	3	6	9	2	5	8	1	4	7	0	3	6	
		/	/	\	\	/	/	\	\	/	/	\	\	
Bight-numbers ⟵		6	3	0	7	4	1	8	5	2	9	6	3	0
Columns: L ⟵ R	13	12	11	10	9	8	7	6	5	4	3	2	1	0

From this algorithm-diagram we read that the 14th half-cycle ($i = 6$) for example, which runs from lower-right to upper-left, intersects the "Standing End" on the Column-numbers 2 and 12, hence its half-cycle algorithm becomes:

$$(1,s)2o - u - o - 2u - o - (1,s)2u\,.$$

The half-cycle algorithm for the 10th half-cycle ($i = 4$) will be:

$$o - (2,s)3u - o - u\,.$$

Note that it intersects the "Standing End" on Column-number 8.

The complete algorithm-table will be as follows:

1.		L ⟶ R :	free run .
2.	$(i = 0)$	R ⟶ L :	$(s)o$.
3.	$(i = 0)$	L ⟶ R :	o .
4.	$(i = 1)$	R ⟶ L :	$(s)u - o$.
5.	$(i = 1)$	L ⟶ R :	$u - o$.
6.	$(i = 2)$	R ⟶ L :	$(s,1)2u - o$.
7.	$(i = 2)$	L ⟶ R :	$2u - o$.
8.	$(i = 3)$	R ⟶ L :	$(s)o - 2u - o - (s)u$.
9.	$(i = 3)$	L ⟶ R :	$o - 2u - o - u$.
10.	$(i = 4)$	R ⟶ L :	$o - (2,s)3u - o - u$.
11.	$(i = 4)$	L ⟶ R :	$o - 3u - o - u$.
12.	$(i = 5)$	R ⟶ L :	$o - u - (s)o - 2u - o - u$.
13.	$(i = 5)$	L ⟶ R :	$o - u - o - 2u - o - u$.
14.	$(i = 6)$	R ⟶ L :	$(1,s)2o - u - o - 2u - o - (1,s)2u$.
15.	$(i = 6)$	L ⟶ R :	$2o - u - o - 2u - o - 2u$.
16.	$(i = 7)$	R ⟶ L :	$2o - u - o - 2u - (s,1)2o - 2u$.
17.	$(i = 7)$	L ⟶ R :	$2o - u - o - 2u - 2o - 2u$.
18.	$(i = 8)$	R ⟶ L :	$2o - u - (1,s)2o - 2u - 2o - 2u$.
19.	$(i = 8)$	L ⟶ R :	$2o - u - 2o - 2u - 2o - 2u$.
20.	$(i = 9)$	R ⟶ L :	$2o - (s,1)2u - 2o - 2u - 2o - 2u$.

Braiding a Column-coded Regular Knot by starting at the centre of the string-length does not affect the Column-numbers where an *even* half-cycle intersects the "Standing End" of the phase in question (recall that the second phase has a different "Standing End" from the first phase).

Thus, for example, if the above knot is braided by starting at the centre of its string-length, and the first phase of braiding is "upwards" with its first half-cycle beginning at a left-hand bight, then the associated algorithm-table is as follows:

	1.		L ⟶ R :	free run .
	2.	$(i=0)$	R ⟶ L :	$(s)o$.
upwards	3.	$(i=0)$	L ⟶ R :	o .
	4.	$(i=1)$	R ⟶ L :	$(s)u-o$.
	5.	$(i=1)$	L ⟶ R :	$u-o$.
	6.	$(i=2)$	R ⟶ L :	$(s,1)2u-o$.
	7.	$(i=2)$	L ⟶ R :	$2u-o$.
	8.	$(i=3)$	R ⟶ L :	$(s)o-2u-o-(s)u$.
	9.	$(i=3)$	L ⟶ R :	$o-2u-o-u$.
	10.	$(i=4)$	R ⟶ L :	$o-(2,s)3u-o-u$.
	11.	$(i=4)$	L ⟶ R :	$u-3o-u-o$.
	12.	$(i=5)$	R ⟶ L :	$u-o-(s)u-2o-u-o$.
downwards	13.	$(i=5)$	L ⟶ R :	$u-o-u-2o-u-o$.
	14.	$(i=6)$	R ⟶ L :	$(1,s)2u-o-u-2o-u-(1,s)2o$.
	15.	$(i=6)$	L ⟶ R :	$2u-o-u-2o-u-2o$.
	16.	$(i=7)$	R ⟶ L :	$2u-o-u-2o-(s,1)2u-2o$.
	17.	$(i=7)$	L ⟶ R :	$2u-o-u-2o-2u-2o$.
	18.	$(i=8)$	R ⟶ L :	$2u-o-(1,s)2u-2o-2u-2o$.
	19.	$(i=8)$	L ⟶ R :	$2u-o-2u-2o-2u-2o$.
	20.	$(i=9)$	R ⟶ L :	$2u-(s,1)2o-2u-2o-2u-2o$.

Braiding by Weaving-pattern

We have remarked earlier that braiding a knot by means of its weaving-pattern is not a recommendable method, for the reasons quoted. However, there will always be braiders who have no desire to produce original or varied work, some of whom will swear by the "weaving-pattern" method of braiding. Needless to say, these braiders will never make the grade of braiding-artisan.

We shall first discuss the most simple form of "weaving-pattern" braiding, whereby each successive *odd* half-cycle is adjacent to its immediately preceeding *odd* half-cycle (and hence each successive *even* half-cycle is also adjacent to its immediately preceeding *even* half-cycle).

This condition can only occur when the value of $|p|_b$ is either equal to 1 or equal to $(b-1)$. It explains the reason why some braiders never progress beyond braiding such dimensioned Regular Knots. Before our braiding methods were developed, it were undoubtedly the easiest of regular knots to braid; but with our braiding methods the braiding of any other dimensioned Regular Knot is just as easy. Hence there is no need to restrict oneself to such common-place knots, which are often dimension-wise, not the best for many applications.

We shall discuss below an example of a Regular Column-coded Knot for each of the above indicated cases (that is for $|p|_b = 1$ and $|p|_b = b - 1$) respectively.

Example 5 :

Suppose the Regular Column-coded Knot is a 17-part/8-bight 2-pass Gaucho Knot with the following coding (note that $|p|_b = |17|_8 = 1$) :

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
right to left ⟶	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The complementary bight-number scheme of this knot can be drawn up by using the value of $|-p|_b$ or by using the value of Δ^* as discussed earlier. The value of $|-p|_b$ is equal to $|-17|_8 = 7$ and the value of Δ^* is 7 (m^* is -1).

The reader will note that, when $|p|_b = 1$, the value of $|-p|_b$ and the value of Δ^* are both always equal to $(b-1)$; and the value $m^* = -1$.

Hence the complementary bight-number scheme of this knot will be as follows :

0 7 6 5 4 3 2 1
. . . .

We have now obtained all the necessary information which enables us to draw up the algorithm-diagram belonging to the knot of our example. This algorithm-diagram is as follows :

Columns : L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Bight-numbers ⟶	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
Bight-numbers ⟵		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0
Columns : L ⟵ R	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

It is now a simple matter to braid the knot directly from this algorithm-diagram, which in this case is identical with braiding it to its weaving-pattern. When the knot is braided upwards, then, since $|p|_b = 1$, the 3^{rd} half-cycle lies immediately above and adjacent to the 1^{st} half-cycle, the 4^{th} half-cycle lies immediately above and adjacent to the 2^{nd} half-cycle, and so on.

Example 6:

Suppose the Regular Column-coded Knot is a 17-part/9-bight 2-pass Gaucho Knot with the following coding (note that $|p|_b = |17|_9 = 8 = b - 1$):

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
right to left ⟶	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The complementary bight-number scheme of this knot can be drawn up by using the value of $|-p|_b$ or by using the value of Δ^* as discussed earlier. The value of $|-p|_b$ is equal to $|-17|_9 = 1$ and the value of Δ^* is 1 (m^* is 0).

The reader will note that, when $|p|_b = b - 1$, the value of $|-p|_b$ and the value of Δ^* are both always equal to 1; and the value $m^* = 0$.

Hence the complementary bight-number scheme of this knot will be as follows:

0	1	2	3	4	5	6	7	8
.	.	.	.	.	.	.	.	.

We have now obtained all the necessary information which enables us to draw up the algorithm-diagram belonging to the knot of our example. This algorithm-diagram is as follows:

Columns: L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Bight-numbers ⟶	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
Bight-numbers ⟵		7	6	5	4	3	2	1	0	8	7	6	5	4	3	2	1	0
Columns: L ⟵ R	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

It is again a simple matter to braid the knot directly from this algorithm-diagram, which in this case is identical with braiding it to its weaving-pattern. When the knot is braided upwards, then, since $|p|_b = 8 = b - 1$, the 3^{rd} half-cycle lies immediately below and adjacent to the 1^{st} half-cycle, the 4^{th} half-cycle lies immediately below and adjacent to the 2^{nd} half-cycle, and so on.

The weaving-pattern method of braiding cannot only be applied to Column-coded Regular Knots for which $|p|_b = 1$ or $|p|_b = b - 1$, but it can also readily be used with those for which $|2p|_b = 1$ or $|2p|_b = b - 1$; $|3p|_b = 1$ or $|3p|_b = b - 1$; and $|4p|_b = 1$ or $|4p|_b = b - 1$.

We have seen that in the cases $|p|_b = 1$ and $|p|_b = b - 1$, the bights on each bight-boundary are formed as one set by laying down each successive bight adjacent to the already existing ones.

For $|2p|_b = 1$ and $|2p|_b = b - 1$, the bights on each bight-boundary form two sets. For $|2p|_b = 1$, one set consists of the bight-index numbers $0, 1, 2, \cdots, (|p|_b - 1)$ and the other consists of the bight-index numbers $|p|_b, (|p|_b + 1), (|p|_b + 2), \cdots, (b-2), (b-1)$.

These sets are built up alternately, starting with bight-index number 0, the next one laid down is bight-index number $|p|_b$, then follows bight-index number 1 adjacent to bight-index number 0, next comes bight-index number $(|p|_b+1)$ adjacent to bight-index number $|p|_b$, then follows bight-index number 2 adjacent to bight-index number 1, next bight-index number $(|p|_b+2)$ adjacent to bight-index number $(|p|_b+1)$, and so on.

For $|3p|_b = 1$ and $|3p|_b = b-1$, the bights on each bight-boundary are formed in three sets; the first set begins with bight-index number 0, the second set begins with bight-index number $|p|_b$, and the third set begins with bight-index number $|2p|_b$. The three sets are built up sequentially with each new bight in a set adjacent to the previous one of the set.

For $|4p|_b = 1$ and $|4p|_b = b-1$, the bights on each bight-boundary are formed in four sets; the first set begins with bight-index number 0, the second set begins with bight-index number $|p|_b$, the third set begins with bight-index number $|2p|_b$, and the fourth set begins with bight-index number $|3p|_b$. The four sets are again built up sequentially with each new bight in a set adjacent to the previous one of the set.

We shall illustrate the above cases with some examples below.

Example 7:

Suppose the Regular Column-coded Knot is a 17-part/11-bight 2-pass Gaucho Knot with the following coding (note that $|2p|_b = |34|_{11} = 1$):

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
right to left ⟶	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

We obtain: $|-p|_b = |-17|_{11} = 5$, $m^* = -1$, $\Delta^* = 9$. Hence its algorithm-diagram is as follows:

Columns: L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Bight-numbers ⟶	0	9	7	5	3	1	10	8	6	4	2	0	9	7	5	3	1	
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
Bight-numbers ⟵		1	3	5	7	9	0	2	4	6	8	10	1	3	5	7	9	0
Columns: L ⟵ R	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Example 8:

Suppose the Regular Column-coded Knot is a 17-part/7-bight 2-pass Gaucho Knot with the following coding (note that $|2p|_b = |34|_7 = 6 = b-1$):

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
right to left ⟶	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

We obtain: $|-p|_b = |-17|_7 = 4$, $m^* = 1$, $\Delta^* = 2$. Hence its algorithm-diagram is as follows:

Columns: L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Bight-numbers ⟶	0	2	4	6	1	3	5	0	2	4	6	1	3	5	0	2	4	
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
Bight-numbers ⟵		4	2	0	5	3	1	6	4	2	0	5	3	1	6	4	2	0
Columns: L ⟵ R	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Example 9 :

Suppose the Regular Column-coded Knot is a 17-part/10-bight 2-pass Gaucho Knot with the following coding (note that $|3p|_b = |51|_{10} = 6 = 1$):

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
right to left ⟶	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

We obtain : $|-p|_b = |-17|_{10} = 3$, $m^* = -1$, $\Delta^* = 7$. Hence its algorithm-diagram is as follows:

Columns : L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Bight-numbers ⟶	0	7	4	1	8	5	2	9	6	3	0	7	4	1	8	5	2	
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
Bight-numbers ⟵		2	5	8	1	4	7	0	3	6	9	2	5	8	1	4	7	0
Columns : L ⟵ R	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Example 10 :

Suppose the Regular Column-coded Knot is a 17-part/13-bight 2-pass Gaucho Knot with the following coding (note that $|3p|_b = |51|_{13} = 12 = b - 1$):

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
right to left ⟶	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

We obtain : $|-p|_b = |-17|_{13} = 9$, $m^* = 2$, $\Delta^* = 3$. Hence its algorithm-diagram is as follows:

Columns : L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Bight-numbers ⟶	0	3	6	9	12	2	5	8	11	1	4	7	10	0	3	6	9	
		/	/	\	\	/	/	\	\	/	/	\	\	/	/	\	\	
Bight-numbers ⟵		9	6	3	0	10	7	4	1	11	8	5	2	12	9	6	3	0
Columns : L ⟵ R	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Example 11 :

Suppose the Regular Column-coded Knot is a 13-part/17-bight 2-pass Gaucho Knot with the following coding (note that $|4p|_b = |52|_{17} = 1$):

columns numbered left to right ⟶	0	1	2	3	4	5	6	7	8	9	10	11	12	13
		/	/	\	\	/	/	\	\	/	/	\	\	
right to left ⟵	13	12	11	10	9	8	7	6	5	4	3	2	1	0

We obtain : $|-p|_b = |-13|_{17} = 4$, $m^* = -1$, $\Delta^* = 13$. Hence its algorithm-diagram is as follows:

Columns : L ⟶ R	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Bight-numbers ⟶	0	13	9	5	1	14	10	6	2	15	11	7	3	
		/	/	\	\	/	/	\	\	/	/	\	\	
Bight-numbers ⟵		3	7	11	15	2	6	10	14	1	5	9	13	0
Columns : L ⟵ R	13	12	11	10	9	8	7	6	5	4	3	2	1	0